Day 84

Q.Given an undirected graph and an integer M. The task is to determine if the graph can be colored with at most M colors such that no two adjacent vertices of the graph are colored with the same color. Here coloring of a graph means the assignment of colors to all vertices. Print 1 if it is possible to colour vertices and 0 otherwise.

Example 1:

Input:
N = 4
M = 3
E = 5
Edges[] = {(0,1),(1,2),(2,3),(3,0),(0,2)}

Output: 1
Explanation: It is possible to colour the given graph using 3 colours.

Example 2:

Input:
N = 3
M = 2
E = 3

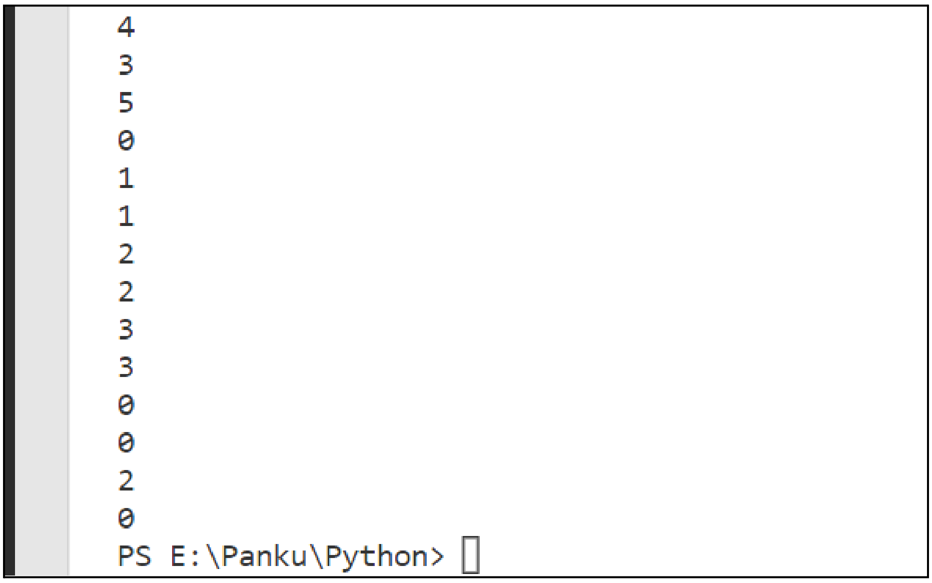
Edges[] = {(0,1),(1,2),(0,2)}

Output: 0

main.py

```
def isSafe(graph, color):
    for i in range(n):
        for j in range(i+1, n):
            if (graph[i][j] and color[j] == color[i]):
                return False
    return True

def graphColoring(graph, m, i, color):
    if (i == n):
        if (isSafe(graph, color)):
            display(color)
            return True
        return False

    for j in range(1, m + 1):
        color[i] = j
        if (graphColoring(graph, m, i + 1, color)):
            return True
        color[i] = 0
    return False

def display(color):
    print("1")

n=int(input())
m=int(input())
e=int(input())
graph=[]

for i in range(n):
    a=[]

    for j in range(n):
        a.append(0)
        graph.append(a)

for i in range(e):
    a=int(input())
    b=int(input())
    graph[a][b]=1
    graph[b][a]=1

color = [0 for i in range(n)]
if (not graphColoring(graph, m, 0, color)):
    print ("0")
```

output



```
4
3
5
0
1
1
2
2
3
3
0
0
2
0
PS E:\Panku\Python> 
```